

REPORT DOCUMENTATION PAGE				Form Approved OMB NO. 0704-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA, 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY) 29-06-2008		2. REPORT TYPE Final Report		3. DATES COVERED (From - To) 18-Jun-2007 - 17-Mar-2008	
4. TITLE AND SUBTITLE Statistical Relational Learning as an Enabling Technology for Data Acquisition and Data Fusion in Heterogeneous Sensor Networks: Final Report				5a. CONTRACT NUMBER W911NF-07-1-0428	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER 611102	
6. AUTHORS David Jacobs, Lise Getoor				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAMES AND ADDRESSES University of Maryland - College Park Office of Research Administration & Advancement University of Maryland, College Park College Park, MD 20742 -5141				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211				10. SPONSOR/MONITOR'S ACRONYM(S) ARO	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) 52970-CI-II.1	
12. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release; Distribution Unlimited					
13. SUPPLEMENTARY NOTES The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.					
14. ABSTRACT Our work has focused on developing new cost sensitive feature acquisition and classification algorithms, mapping these algorithms onto camera networks, and creating a test bed of video data and implemented vision algorithms that we can use to implement these. First, we will describe a new algorithm that we have developed for feature acquisition in Hidden Markov Models					
15. SUBJECT TERMS statistical relational learning, cost-sensitive classification, feature acquisition, sensor networks					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT SAR	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Lise Getoor
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER 301-405-2691

Report Title

Statistical Relational Learning as an Enabling Technology for Data Acquisition and Data Fusion in Heterogeneous Sensor Networks: Final Report

ABSTRACT

Our work has focused on developing new cost sensitive feature acquisition and classification algorithms, mapping these algorithms onto camera networks, and creating a test bed of video data and implemented vision algorithms that we can use to implement these. First, we will describe a new algorithm that we have developed for feature acquisition in Hidden Markov Models (HMMs). This is particularly useful for inference tasks involving video from a single camera, in which the relationship between frames of video can be modeled as a Markov chain. We describe this algorithm in the context of using background subtraction results to identify portions of video that contain a moving object. Next, we will describe new algorithms that apply to general graphical models. These can be tested using existing test sets that are drawn from a range of domains in addition to sensor networks.

List of papers submitted or published that acknowledge ARO support during this reporting period. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Sen, Prithviraj and Getoor, Lise,
"Cost-Sensitive Learning with Conditional Markov Networks",
Data Mining and Knowledge Discovery, Special Issue on Utility Based Data Mining", 2008.

Number of Papers published in peer-reviewed journals: 1.00

(b) Papers published in non-peer-reviewed journals or in conference proceedings (N/A for none)

Number of Papers published in non peer-reviewed journals: 0.00

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts): 0

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Bilgic, Mustafa and Getoor, Lise, "Effective Label Acquisition for Collective Classification", ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008.

Bilgic, Mustafa and Getoor, Lise, "VOILA: Efficient Feature-value Acquisition for Classification", AAAI '07: Proceedings of the 22nd National Conference on Artificial Intelligence, 2007.

(d) Manuscripts

Number of Manuscripts:0.00

Number of Inventions:

Graduate Students

<u>NAME</u>	<u>PERCENT SUPPORTED</u>
Mustafa Bilgic	0.25
Daozheng Chen	0.50
Galileo Namata	0.13
Prithviraj Sen	0.13
FTE Equivalent:	1.01
Total Number:	4

Names of Post Doctorates

<u>NAME</u>	<u>PERCENT SUPPORTED</u>
FTE Equivalent:	
Total Number:	

Names of Faculty Supported

<u>NAME</u>	<u>PERCENT SUPPORTED</u>	National Academy Member
David Jacobs	0.04	No
Lise Getoor	0.04	No
FTE Equivalent:	0.08	
Total Number:	2	

Names of Under Graduate students supported

<u>NAME</u>	<u>PERCENT SUPPORTED</u>
Daozheng Chen	0.50
FTE Equivalent:	0.50
Total Number:	1

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period:	1.00
The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:.....	0.00
The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:.....	1.00
Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):	1.00
Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:.....	0.00
The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense	0.00
The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:	0.00

Names of Personnel receiving masters degrees

NAME

Total Number:

Names of personnel receiving PhDs

NAME

Total Number:

Names of other research staff

NAME

PERCENT SUPPORTED

FTE Equivalent:

Total Number:

Sub Contractors (DD882)

Inventions (DD882)

Statistical Relational Learning as an Enabling Technology for Data Acquisition and Data Fusion in Heterogeneous Sensor Networks: Final Report

Lise Getoor and David Jacobs
University of Maryland, College Park

Contact Information

Prof. Lise Getoor
Computer Science Department
AV Williams Bldg
University of Maryland
College Park, MD 20912
Phone: 301-405-2691
E-mail: getoor@cs.umd.edu

Our work has focused on developing new cost sensitive feature acquisition and classification algorithms, mapping these algorithms onto camera networks, and creating a test bed of video data and implemented vision algorithms that we can use to implement these. First, we will describe a new algorithm that we have developed for feature acquisition in Hidden Markov Models (HMMs). This is particularly useful for inference tasks involving video from a single camera, in which the relationship between frames of video can be modeled as a Markov chain. We will describe this algorithm in the context of using background subtraction results to identify portions of video that contain a moving object. Next, we will describe new algorithms that apply to general graphical models. These can be tested using existing test sets that are drawn from a range of domains in addition to sensor networks.

1 Feature Acquisition within a Single Camera

We have completed a preliminary study aimed at performing cost-sensitive label acquisition for background subtraction in a single video stream. We describe a small set of initial results that we find encouraging.

We consider the problem of using background subtraction to determine whether there is a moving object in a video (see Figure 1). Objects are detected by first performing background subtraction, and then using the size of the largest connected component of foreground pixels as a feature. Generally, this is small when noise causes scattered foreground pixels, and larger when there is a moving object. We can integrate information temporally using a Hidden Markov Model with two states that reflect the presence or absence of a moving object. Each state gives rise to a different distribution for the size of the largest connected component.

In this simple setting we can examine the problem of performing label acquisition to control the use of cheap and expensive background subtraction algorithms. For a cheap algorithm, we simply threshold the difference between each frame and the previous one, marking pixels with large intensity differences as foreground. As a more expensive algorithm we use a mixture of Gaussians to model the background, and mark foreground pixels that are unlikely to be background according to this model [7]. The question we face is whether we can run the cheap algorithm on all frames but use the expensive algorithm sparingly, and still achieve accuracy similar to that obtained when we run the both algorithms everywhere.

If we idealize the situation slightly, and assume that the expensive algorithm always produces an accurate result, then [3] show that we can use dynamic programming (DP) to find the optimal locations at which to apply the expensive algorithm. Unfortunately, given n nodes in an HMM, their algorithm requires $O(n^5)$ computation. This algorithm is suitable for an HMM that has few nodes, when obtaining new measurements is very expensive, but not for a video containing thousands of frames. However, we have experimented with snippets of video containing dozens of frames. For these we can use the DP algorithm to select frames for expensive background subtraction, producing results that are twice as accurate as when we simply apply expensive processing to the same number of frames selected uniformly in the sequence. This is encouraging, but significantly



Figure 1: Our background subtraction experiments use video from outdoor scenes, as shown above.

understates the potential of intelligent label acquisition.

If we divide a video into short snippets, the primary potential for intelligent processing is probably in deciding which snippets deserve a lot of extra attention and which do not. We have therefore devised a new algorithm that applies expensive processing to every 50th frame and then runs DP on each snippet. We have developed a novel method to combine these results efficiently and optimally to tell us in which snippets we should apply expensive processing, and where in these snippets to process. This gives us an algorithm that, as overhead, requires application of expensive processing to a constant fraction of the frames, but in return reduces computation time from $O(n^5)$ to $O(n)$, yielding an algorithm suitable for long video streams.

In section 1.1, we describe how we map a Hidden Markov Model to the problem of identifying interesting events in a single video stream. In section 1.2, we formulate the problem more concretely. In section 1.3, we describe a dynamic programming algorithm to solve the problem optimally. In section 1.4, we discuss how we modify the algorithm in section 1.3 to meet our need. In section 1.5, we describe and discuss the simple experiment we have performed so far.

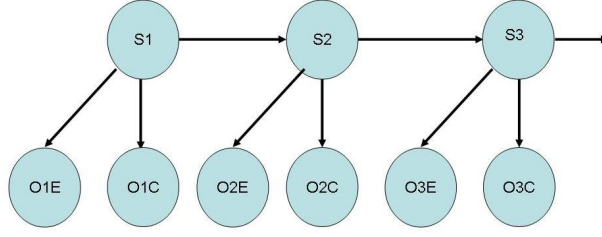


Figure 2: The HMM model

1.1 Hidden Markov Model and Probabilistic Inference

To start, we notice that whether or not a frame of video is interesting is closely related to whether its neighboring frames are interesting. This suggests that we map the problem of identifying significant sequences of video onto a Hidden Markov Model (HMM), which will enable us to do probabilistic inference. In our model, each frame in the sequence is associated with a state variable. The state is either interesting or not interesting. The observation emitted by the state variable corresponds to the features extracted from the frame. For simplicity, one cheap feature and one expensive feature are considered for now. The structure of this model is shown in figure 2. Variables starting with label “S” are state variables, and variables starting with “O” are observations, where cheap features are labeled “C” and expensive features are labeled “E”. After learning the parameters of this model, we can use a standard inference algorithm [4] to determine the state of each frame based on their features. This sets up the problem of determining from which frames we should extract expensive features. The following section gives a more concrete formulation of this problem.

1.2 Problem Formulation

The following problem formulation uses the same formulation described in section 2 in [3]. First, we want to model the fact that observations are informative. We consider a sequence of state variables $S = X_1, \dots, X_n$ in the HMM, and we define a class of local reward functions R on the marginal probability distributions of the variables. The local reward can be evaluated using probabilistic inference techniques, and the total reward will then be the sum of all local rewards. To define a local reward, we use a functional on the max-marginal $P^{max}(X_j|O)$ for classification purpose,

$$R_j(X_j|O) = R_j(P^{max}(X_j|O)) = \sum_o P(o)[P^{max}(x_j^*|o) - P^{max}(\bar{x}_j|o)],$$

where O is the set of observed state variables, o is their value, x_j is the value of X_j , $x_j^* = \operatorname{argmax}_{x_j} P^{max}(x_j|o)$, and $\bar{x}_j = \operatorname{argmax}_{x_j \neq x_j^*} P^{max}(x_j|o)$. Second, we want to capture the constraint that observations are expensive. This can mean that each observation X_j has an associated positive penalty C_j that effectively decreases the reward. Third, it is also possible to define a budget B for selecting observations, where each one is associated with an integer cost β_j . Finally, our formulation of the optimization

problem is

Maximize

$$J(O) = R(O) - C(O) = \sum_j R_j(X_j|O) - \sum_i C_i$$

Subject to

$$\sum_i \beta_i < B$$

where j is the index over all the state variables S , i is the index over observed state variables O , and B is the total amount of budget for the subsequence.

In addition, the property that our the state variables in our HMM form a Markov chain simplifies this local reward. By conditional independence properties in a graphical model, the local reward is simply $R(X_j|O) = R(X_j|X_j)$ in the case that $X_j \in O$. In the case that X_j is not in O , we have $R(X_j|O) = R(X_j|O_j)$, where O_j is the subset of O containing the closest ancestor and descendant of X_j in O . This local reward simplification plays a key role in the optimization algorithm in section 1.3.

1.3 Conditional Plan

To get the set of observations for the problem, we need to specify a query policy. We consider the following conditional policy: we sequentially observe the state variable in the HMM, pay the penalty, and depending on the observed values, select the next query as long as our budget suffices. Putting this policy into the optimization problem above, our goal is to find a plan with the highest expected reward, where, for each possible sequence of observations, the budget B is not exceeded. We call such a observation plan the conditional plan [3]. To solve this problem, the objective function J to be maximized is defined recursively:

The base case is defined on budget 0:

$$J(O = o; 0) = \sum_{X_j \in S} R_j(X_j|O = o) - C(O)$$

The recursive case is defined on budget limited to k :

$$J(O = o; k) = \max\{J(O = o; 0), \max_{X_j \text{ not in } O} \sum_y P(X_j = y|O = o) J(O = o, X_j = y; k - \beta_j)\}$$

The optimal plan has reward $J(\emptyset; B)$. [3] provides a dynamic programming algorithm on a sub-chain of state variables X_a, \dots, X_b . The base case is defined on budget 0:

$$J_{a:b}(x_a, x_b; 0) = \sum_{j=a+1}^{b-1} R_j(X_j|X_a = x_a, X_b = x_b),$$

where, $J_{a:b}(x_a, x_b; 0)$ means the optimal plan reward given $X_a = x_a$ and $X_b = x_b$ with total budget 0. The recursive case defines $J_{a:b}(x_a, x_b; k)$, in which the total budget is k :

$$J_{a:b}(x_a, x_b; k) = \max\{J_{a:b}(x_a, x_b; 0), \max_{a < j < b} \{-C_j + \sum_{x_j} P(X_j = x_j|X_a = x_a, X_b = x_b)\}$$

$$R_j(X_j|X_j = x_j) + \max_{0 \leq l \leq k - \beta_j} [J_{a:j}(x_a, x_j; l) + J_{j:b}(x_j, x_b; k - l - \beta_j)] \}} \}$$

where it iterates through the possible split points j , such that $a < j < b$, and consider all possible budget allocation l and $k - l - \beta_j$ for sub-chains to find maximum reward. Using the property that reward is decomposable along chain, the optimal reward $J(\emptyset; B)$ is obtained by $J_{0:n+1}(\emptyset; B)$. The state variables X_0 and X_{n+1} referred in $J_{0:n+1}(\emptyset; B)$ are two dummy variables with $x_0 = 1, x_{n+1} = 1$, and $R_0 = C_0 = \beta_0 = R_{n+1} = C_{n+1} = \beta_{n+1} = 0$. After this, the observation plan can be obtained by backtracking observation at each step of the resulting conditional plan. For details of this algorithm, please refer to section 5 in [3].

1.4 Subsections and Convexity of Reward

1.4.1 Subsections

Using the above dynamic programming algorithm, an optimal conditional plan can be obtained. However, one issue with this approach is that the running time of this algorithm is $d^3 B^2 (1/6n^3 + O(n^2))$, where n is the number of state variables in HMM and d is the maximum domain size of the state variables X_1, \dots, X_n ($d = 2$ in our case). This algorithm is therefore only suitable for situations in which n is a fairly small value, such as $n = 50$ or 100 . In the case of video sequences, n may be 100,000 or greater, requiring an algorithm that is approximately linear in n .

To solve this problem, we consider the Markov properties of our HMM, which says that the value of a state variable only depends on the values of its nearby state variables. So to predict the state correctly, we may only need some expensive features in some nearby positions rather from the whole sequence. Inspired by this, we divide the whole sequence into subsections, and run the dynamic programming algorithm in each subsection to select expensive features in each subsection. Given subsequences of constant length, this can be done in $O(n)$ time. The key question is then to allocate the available budget for processing between all of these subsequences.

1.4.2 Convexity of Reward

To run the dynamic programming algorithm, we may need a budget for each subsection. A simple scheme is to uniformly allocate budget to each subsection using the total budget. However, some sections may need more expensive features to eliminate false positives or false alarms. At the same time, many other subsequences will require few, if any, additional expensive features because they are unlikely to contain any events of interest.

We are therefore left with the following problem. For each subsequence, we can determine the optimal feature acquisition plan for every possible budget, and the expected reward of each budget. How do we allocate a single budget among all these subsequences to maximize our expected reward? This problem has a simple solution for the special case in which the expected reward from acquiring expensive features is monotonically increasing and convex with the number of features acquired. That is, the reward must follow a law of diminishing returns in which increasing the budget increases the reward more and more slowly. In this case, we can allocate our budget

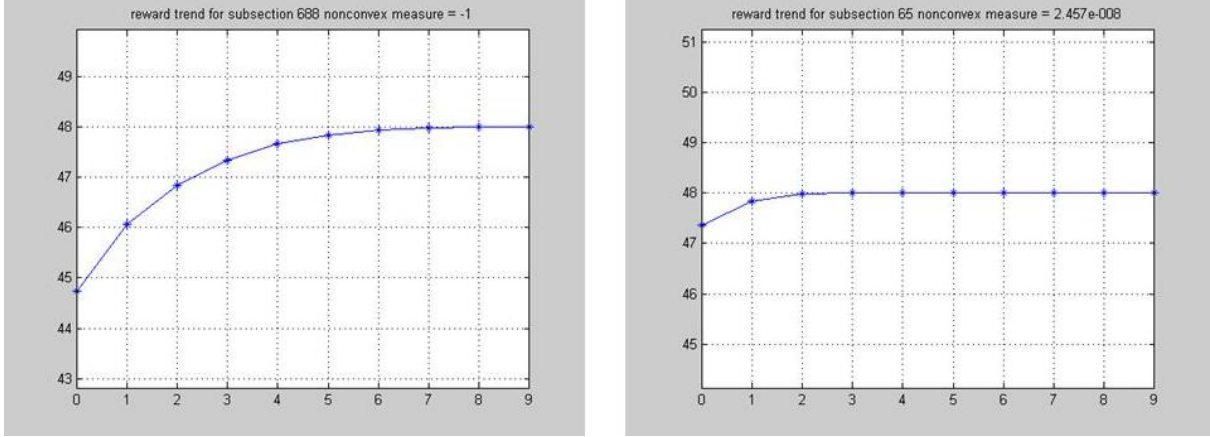


Figure 3: A convex curve (left) and a non-convex curve (right)

optimally by assigning available resources incrementally to whichever subsequence will benefit most from them. We stress that this incremental approach assigns the budget optimally; within each subsequence the dynamic programming algorithm is used to then determine the optimal set of observations for the assigned budget.

Empirically, we do find that the rewards of feature acquisition are convex for our problem. To test this we obtained a sequence of 64,000 consecutive frames from a video viewing the bike track around a lake. We divided the sequence into subsections, each of which contains fifty consecutive frames. A slightly modified version of the dynamic programming algorithm (please refer to the experiment section below) with budget from 0 to 9 was run on each subsection. Then we plot a curve for each subsection describing the changes of optimal plan reward as budget changes, and we call it reward-budget (RB) curve. We discover that the RB curve is always either strictly convex (see the left side in Figure 3) or non-convex by a tiny amount (see the right side of Figure 3).

We can therefore use the following method to maximize the sum of reward increments for each subsection. Let N be the maximum budget in the RB curves for each subsection. We can compute the reward increments in each subsection for budget increments from zero to one, from one to two, ..., and from $N - 1$ to N . We sort all these increments, and the budget for each subsection is determined by the number of increments it has in the sorted top B increment list. We call this method batch allocation of budget.

We can further improve on this, and avoid wasted computation with the following algorithm:

1. Initialize the budget of each subsection to be zero;
2. Compute the increment of reward for each subsection for increasing the budget from zero to one;
3. Select the highest increment of reward, and add one more budget to its corresponding subsection Q ;

4. If the total budget for the whole sequence has been used up, terminate and use the current subsection budget allocation as the final subsection budget allocation;
5. Compute the reward increment for subsection Q when its current budget is incremented by one, and use it to replace the current reward increment for this subsection. Go back to step 3;

We call this algorithm dynamic allocation of budget. We give a proof of this algorithm below to show that it gives an optimal budget allocation to maximize the increment of reward.

Problem: Given a total budget B , and let ΔR_j^i , where $i, j \in Z^+$, be the increment of reward of optimal plan for subsection j when budget going from $i-1$ to i . Then by the convexity assumption, we know that $\Delta R_j^i \geq \Delta R_j^{i+1}$ for every possible i and j . Let α_j be the budget for subsection j , and $R_j^{\alpha_j}$ be the total increment of reward for subsection j with budget α_j . Then we have

$$R_j^{\alpha_j} = \begin{cases} 0 & \text{if } \alpha_j = 0 \\ \sum_{i=1}^{\alpha_j} \Delta R_j^i & \text{if } \alpha_j > 0 \end{cases}$$

show that the dynamic allocation of budget algorithm minimizes $\sum_j^N R_j^{\alpha_j}$, where N is the total number of subsections.

Proof: First, we observe that if $\Delta R_n^p > \Delta R_m^q$, then ΔR_n^p will be selected before ΔR_m^q is selected. To show this is true, suppose ΔR_m^q is selected first, then it must be greater than or equal to every reward increment for every subsection at that particular moment, including the one for subsection n . Let the reward increment be ΔR_n^w . Since $\Delta R_n^p > \Delta R_m^q$, then $\Delta R_n^p > \Delta R_n^w$. By convexity assumption, we know that $p < w$. Then it must be that ΔR_n^p is selected before ΔR_n^w , and as a result before ΔR_m^q . But this creates a contradiction with our original assumption. This observation is denoted as lemma 1.

Second, let the the optimal budget allocation be the set $\{\beta_j\}$, where β_j be the budget for subsection j . And let $\{l\}$ be the set of indices for those subsections whose budget is zero. Then by the definition of $R_j^{\beta_j}$, we have $\sum_j^N R_j^{\beta_j} = \sum_l \sum_{k=1}^{\beta_l} \Delta R_l^k$. Sorting ΔR_l^k for all possible l and k , we can obtain $\sum_j^N R_j^{\beta_j} = \sum_{b=1}^B \Delta T_b$, where each ΔT_b is corresponding to a ΔR_l^k , and $\Delta T_b \geq \Delta T_{b-1}$ for every possible b . Also, let ΔP_b be the b th reward increment selected in the algorithm, and γ_j be the budget for subsection j computed by the algorithm. Then we can see that the total increment of reward by the algorithm $\sum_j^N R_j^{\gamma_j} = \sum_{b=1}^B \Delta P_b$. The observation that $\Delta T_b \leq \Delta P_b$ for every possible b is shown below. This observation is denoted as lemma 2.

Suppose that $\Delta T_b > \Delta P_b$ for some b . First, we know that ΔT_b is equal to some reward increment ΔR_n^p and the same fact holds for ΔP_b . Also, $\Delta T_c \geq \Delta T_b$ for each possible c , such that $c < b$, so $\Delta T_c > \Delta P_b$ for each c . Then by lemma 1, we know that ΔT_c for each possible c and ΔT_b will be selected before ΔP_b in the algorithm, and totally, there are b of them. However, there are only $b-1$ increment of reward is selected before ΔP_b is selected by the definition of ΔP_b . So this creates a contradiction.

Finally, by lemma 2, we know that $\sum_j^N R_j^{\gamma_j} = \sum_{b=1}^B \Delta P_b \geq \sum_{b=1}^B \Delta T_b = \sum_j^N R_j^{\beta_j}$. Also since $\{\beta_j\}$ is the set of optimal budget allocation, we have $\sum_j^N R_j^{\gamma_j} \leq \sum_j^N R_j^{\beta_j}$. As a result, $\sum_j^N R_j^{\gamma_j} = \sum_j^N R_j^{\beta_j}$.

The proof is completed.

1.5 Experiments

Currently, only a simple experiment is performed on the 64000 frames mentioned in the previous section. The real state of each frame is labeled by hand. The features we use are the size of the largest connected foreground component from two background subtraction algorithms: Frame Difference (FD) [2] and Adaptive Gaussian Mixture Model (AGMM) [7]. After the background subtraction, some morphological operations are performed before computing the size. Since AGMM is more accurate than FD in detecting the foreground region and more time-consuming, we consider features from AGMM as expensive features and features from FD as cheap features. To do inference on HMM, we extract the cheap feature at every frame and use the dynamic programming method with subsection to determine on which frames we should sample expensive features. The total 64000 frames are divided into subsections, each of which consists of 50 consecutive frames. In [3], it mentions using dummy state variables with known value at the beginning and end of the sequence before running the dynamic programming algorithm. However, we directly use the real state of the first and last frame in each subsection to run the algorithm. The local reward R_j is computed based on the marginal probability produced by cheap features, penalty C_j associated with a expensive feature observation is always zero, and the observation cost for an expensive feature β_j is always one. The budget for each subsection determined using batch allocation method, and the RD curves for each subsection are computed up to budget 9. To make the learned parameters of the HMM confirm with distribution in the testing data, we do testing and training based on the same data. The inference error rates purely using expensive or cheap features are shown in Table 1. The error rates for our sampling method and uniform sampling method under different total budgets for the whole sequence are shown in Table 2.

Table 1: error rates purely using one kind of features

Cheap Features	Expensive Features
0.0066	0.0018

Table 2: error rates under different budgets for expensive features

Budget	Our Sampling	Uniform Sampling
1850	0.004	0.0059
910	0.004	0.0064
770	0.004	0.0065
570	0.0042	0.0065

From these tables, we can see that by using a few expensive feature samples, our sampling algorithm can achieve better inference accuracy than uniform sampling. Also, compared with the error rate of purely using expensive feature, the error rate is still a bit high using the budget we have tried so far.

1.6 Summary

The main purpose of this simple experiment has been to make our proposed work more concrete. Even for a simple motion-detection task, we see there is a significant trade-off available between accuracy and the amount of processing we perform. This trade-off becomes continuous if we process all frames cheaply and some frames more accurately; by combining all the results using a graphical model, expensive processing in some frames helps us to better analyze all the frames. Given a fixed budget of accurate processing, making the most of this budget is a problem of label acquisition. Performing effective label acquisition in a long video sequence is still an open problem, even when we use a simple HMM as our graphical model. However, we have made significant progress on this problem. Our proposed work aims to extend this simple example to more complex vision tasks and richer graphical models.

2 Extension to More Complex Models

The method we described in the previous section is applicable to HMMs, which is quite reasonable for a single video sequence. However, when we have a network of cameras, we need to integrate information from multiple sources and we need to make simultaneous decisions about whether there was an object in any of the cameras and if so, in which ones. In such situations, we probably need more complex graphical models, such as arbitrary Markov Random Fields. When the underlying graphical model structure is irregular, the algorithms we described in the previous section are not applicable anymore. To address this issue, we have developed a new algorithm, called Reflect and Correct (RAC) [1].

RAC is an iterative algorithm that first finds the locations where an incorrect decision is made and then acquires more information in those locations. The key element of RAC is the question of how to figure out if a frame is misclassified. We answer this question by using a local classifier that makes independent decisions for each frame and by comparing its label estimates with the estimates of the graphical model. We construct some features using the comparison of the estimates and fit a classifier that can predict if a frame is misclassified.

Our preliminary experiments with synthetic datasets and publication datasets are very encouraging. RAC significantly outperformed (in terms of accuracy of the labeling) a viral marketing based strategy [6] and previous approaches that are based on network structural properties such as network clustering and node degree [5].

References

- [1] Mustafa Bilgic and Lise Getoor. Effective label acquisition for collective classification. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008.
- [2] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, August 2002.

- [3] A. Krause and C. Guestrin. Optimal nonmyopic value of information in graphical models: Efficient algorithms and theoretical limits. In *The Nineteenth International Joint Conference on Artificial Intelligence*, 2005.
- [4] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1989.
- [5] Matthew Rattigan, Marc Maier, and David Jensen. Exploiting network structure for active inference in collective classification. In *ICDM Workshop on Mining Graphs and Complex Structures (MGCS)*, 2007.
- [6] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.
- [7] Z.Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the International Conference on Pattern Recognition*, 2004.